数学启发算法求解单源设施区位问题

孔云峰[1,2]

1 河南大学黄河中下游数字地理技术教育部重点实验室，河南 开封，475000；2 河南大学环境与规划学院，河南 开封，475000

摘要：本文提出了一个数学启发算法求解单源设施区位问题(SSCFLP)。该算法从一个初始解开始，迭代地对当前解进行超大邻域搜索改进，直到若干次尝试不能该改进当前解为止。算法中，初始解使用拉格朗日松弛启发算法，超大邻域搜索采用子问题数学模型精确求解。算法设计的要点在于超大邻域的选择，既不能太大造成子问题求解困难，也不能太小造成当前解难以改进。使用 5 组 272 个基准测试案例进行算法测试，共发现 191 个案例的最优解，更新 36 个案例的已知最好解，表明数学启发算法性能优异。与近年的代表性算法(如割平面、核搜索、超图多交换启发、廊道方法)进行比较，在大规模案例上，本文算法无论求解质量还是计算时间均具有显著的优势。

关键词：单源设施区位问题; 数学启发算法; 超大规模邻域搜索; 子问题

引言

设施区位问题(FLP)是一大类寻找最优设施位置的问题。该问题广泛应用于学校、医疗、公安、养老、应急、物流等公共或商业设施选址决策。根据应用场景的不同，FLP 类型众多：连续空间区位或离散区位、有无设施容量限制、需求是否允许拆分指派、是否考虑设施成本、是否指定设施数量等，规划目标划分为效率目标、公平目标等。根据设施服务与需求是否存在不确定性，又可将 FLP 划分为确定性 FLP 和随机性 FLP。其中，单源 CFLP 问题(SSCFLP)是需求不允许拆分指派的、有容量约束的、顾及设施成本的离散区位问题，求解难度较高，也是近年关注度最高的区位问题之一。

令集合I为潜在设施区位，集合J为客户区位，设施i ($i\in I$)有最大服务容量s_i和固定建设成本为f_i，客户j ($j\in J$)有服务需求d_i，被设施i服务的费用为c_{ij}。定义布尔型决策变量y_i为是否在区位i建设施，布尔型决策变量x_{ij}为是否指派设施i服务客户j，SSCFLP数学模型如下：

$$\text{Minimize} \quad \sum_{i\in I} f_i + \sum_{i\in I}\sum_{j\in J} c_{ij}x_{ij} \tag{1}$$

$$\text{Subject to:} \quad \sum_{i\in I} x_{ij} = 1, \forall j \in J \tag{2}$$

$$\sum_{j\in J} d_j x_{ij} \le s_i y_i, \forall i \in I \tag{3}$$

$$y_i = \{0,1\}, \forall i \in I \tag{4}$$

项目资助：国家自然科学基金项目(41871307)

通讯作者：孔云峰，博士，教授，主要从事空间分析、空间优化等研究。yfkong@henu.edu.cn

$$x_{ij} = \{0,1\}, \forall i \in I, j \in J \quad (5)$$

模型中，目标函数(1)最小化设施开设成本和获取服务成本。约束(2)确保每个客户由唯一的设施提供服务；约束(3)要求指派给每个设施的客户需求量不超过其最大服务容量。约束(4)和(5)定义两类布尔型决策变量。

求解 SSCFLP 问题的方法划分为两大类：精确算法和启发式算法(Basu et al. 2015, Ulukan & Demircioğlu 2015)。精确算法包括分枝定界(Neebe & Rao 1983，Holmberg et al. 1999)、列生成(Díaz & Fernández 2002)、Cut-and-Solve (Yang et al. 2012, Gadegaard et al. 2018)等。因 SSCFLP 问题具有极高的计算复杂度，精确算法难以求解规模较大的案例。

与精确算法相比，求解 SSCFLP 的启发式算法相当多样。基于拉格朗日松弛的启发式算法(LH)应用最广泛(Barcelo & Casanova 1984; Klincewicz & Luss 1986; Beasley 1993; Sridharan 1993; Agar & Salhi 1998; Hindi & Pienkosz 1999; Cortinhal & Captivo 2003)。该方法将指派约束或容量约束整合到目标函数中，然后迭代执行以下步骤：①对偶问题求解，更新原问题下界；②利用对偶问题解构造原问题可行解，更新原问题上界；③次梯度方法更新乘子。基于提供的问题下界，LH 算法常常和其他算法整合使用。例如，Holmberg et al. (1999)将 LH 算法与分枝定界精确算法框架整合，Chen & Ting (2008)将 LH 算法与蚁群系统混合。其他启发式算法包括：禁忌搜索(Filho & Galvao1998; Delmaire et al. 1999; Cortinhal & Captivo 2003)、超大规模邻域搜索(VLNS) (Ahuja et al. 2004; Tran et al. 2017)、分散搜索(Contreras & Diaz, 2008)、核搜索(Kernel Search)(Guastaroba & Speranza 2014)、“廊道”方法(Corridor Method)(Caserta & Voß, 2020)等。

近十余年来，求解 SSCFLP 的代表性算法如下：核搜索 (Guastaroba & Speranza 2014)、割平面(Yang et al. 2012; Gadegaard et al. 2018)、超大邻域搜索(Tran et al. 2017)和数学启发(Caserta & Voß, 2020)。这些算法在求解质量方面具有竞争力，也测试了多组案例，验证了算法的通用性。因这些算法计算时间偏长，Oliveira et al.(2020) 设计了松弛自适应记忆规划(RAMP)方法，大幅降低计算时间，获得较为满意的结果。

本文提出一个数学启发式算法求解 SSCFLP 案例。五组 272 个基准测试案例的实验表明，数学启发式算法在求解大规模 SSCFLP 案例中具有竞争优势：计算结果优于目前公认最好的几个算法，且求解效率显著提升。

1 数学启发算法

邻域搜索用于渐进改进当前解，是众多优化问题启发算法的关键步骤。区位问题中，设施的变动必然会引起需求指派变化，使得邻域搜索较为复杂。因此，区位问题算法设计的一个核心问题是：如何有效地移动设施并更新指派，寻找到可行的、更节约的邻域解。考虑到 SSCFLP 存在设施容量约束，涉及较多设施的超大邻域搜索才有较高可能性改进当前解。然而，SSCFLP 的大邻域结构定义复杂，邻域搜索的计算复杂度也很高(Ahuja et al. 2004; Tran et al. 2017)，均存在指数级数量的客户移动移动和设施移动。

本文拟采用数学模型求解 SSCFLP 的超大邻域。算法原理如下：①使用 LH 方法(Holmberg et al. 1999)获得问题初始解；②从当前解中随机选择一个超大邻域，获得邻域

内需求点、当前设施和候选设施集合；③构造子问题模型并求解，更新设施集合和需求指派；④重复执行步骤②和③，直到若干次尝试不能更新当前解。步骤②构造一个大邻域，步骤③用于发现邻域内的最优解。基于这一思路，数学启发算法示意如下：

```
参数:连续未更新最好解循环数(mloops)
1. s_best = LagrangianHeuristic(psize);
2. notImpr=0;
3. While notImpr < mloops:
4.       Select a large neighborhood I* and J* from solution s;
5.       s*= SolveSubproblem(I*, J*);
6.       s'=UpdateSolution(s,s*)
7.       If f(s')<f(s_best): s_best=s', notImpr=0;
8.           else: notImpr+=1;
9. Output s_best.
```

以上算法中，步骤(1)使用拉格朗日松弛启发方法(Holmberg et al. 1999)获得初始解。模型步骤(4)从当前解中随机选择一个超大邻域，包括设施子集 I^*和需求子集 J^*。步骤(5)使用设施子集和需求子集构造子问题模型，使用 MIP 优化器求解，获得局部解 s^*。步骤(6)使用局部解更新当前解，获得新解 s'。步骤(7)尝试更新最好解。当若干次循环未更新最好解，算法终止。

从当前解中选择一个超大邻域(步骤 4)是算法设计的要点。邻域选择步骤如下：①随机选择一个需求点，以该点为中心，从当前设施集合中自近而远地选择 K 个；②指派给 K 个设施的需求点构成需求点子集 J^*；③子集 J^*中每个需求点最近的设施以及已选择的 K 个设施构成子集 I^*。以上步骤所选择邻域的大小受 K 值影响，K 值取值范围为$[K_{min}, K_{max}]$，其中，$K_{min}=\min(L/2,7)$，$K_{max}=\min(L,10)$，L 为当前设施数量。当 $L\le7$ 时，K 处于区间$[L/2,L]$；当 $8\le L\le10$ 时，K 处于区间$[L/2,L]$；当 $11\le L\le13$ 时，K 处于区间$[L/2,10]$；当 $L\ge14$ 时，K 处于区间[7,10]。若邻域太小，设施区位往往难以更新；若邻域过大，模型计算效率低。

以上邻域选择 K 的取值可以保证邻域足够大，但可能会使模型计算效率低下。为此，当集合 I^*的规模偏大时，需要精简一部分候选设施，再进行子问题建模和求解。精简候选设施采用以下步骤：第一，若当前解目标值小于某个设施对应的排除阈值，则这个设施不在最优解中。设施排除方法来自文献(Holmberg et al. 1999)提供的 LH 算法，即针对拉格朗日乘子，计算对偶问题中每个设施子问题目标值和对偶问题目标值，若某个设施子问题目标值与对偶问题目标值之和大于原问题上界，则该设施不在最优解中。本文 LH 算法记录并更新每个设施对应的被排除目标值阈值，随着解的不断改进，可能有越来越多的设施被排除在最优解之外。第二，排除部分设施后，如果设施仍然比较多，先选择 K 个已使用设施，再从其余设施中随机选择 K 个设施。

2 算法测试结果

2.1 测试案例

本文收集 5 个 SSCFLP 基准案例集，包含 272 个实例。案例集名称、来源、分组及规模汇总于表 1。案例集 OR-Library、Holmberg 、Yang 和 Tebd1 数据见网站 https://or-brescia. unibs.it/instances/instances_sscflp；案例集 TB4 见 https://github.com /SuneGadegaard /SSCFLPsolver。表中，每个案例集划分为若干分组，分组案例名称列在分组后面括号内；$|I|$ 和 $|J|$ 分表表示候选设施数量和需求点数量。这些案例集在供需规模、分布、比例等方面均具有代表性，广泛应用于区位问题算法测试。

表 1 基准测试案例集

案例集	分组(案例名称)	案例数量	$\|I\|$	$\|J\|$
OR-Lib	OR1 (cap61-cap74)	8	16	50
(Ahuja et al. 2004)	OR2 (cap91-cap104)	8	25	50
	OR3 (cap121-cap134)	8	50	50
	OR4 (capax, capbx, capcx)	12	100	1000
Holmberg	H1 (p1-p12)	12	10	50
(Holmberg et al. 1999)	H2 (p13-p24)	12	20	50
	H3 (p25-p40)	16	30	150
	H4 (p41-p55)	15	10-30	70-100
	H5 (p56-p71)	16	30	200
Yang	Y1 (30_200_x)	5	30	200
(Yang et al. 2012)	Y2 (60_200_x)	5	60	200
	Y3 (60_300_x)	5	60	300
	Y4 (80_400_x)	5	80	400
TB4	G1 (50_100_x_x)	15	50	100
(Gadegaard et al. 2018)	G2 (50_200_x_x)	15	50	200
	G3 (60_300_x_x)	15	60	300
Tebd1	T1 (i300_x)	20	300	300
(Avella and Boccia 2009)	T2 (i3001500_x)	20	300	1500
	T3 (i500_x)	20	500	500
	T4 (i700_x)	20	700	700
	T5 (i1000_x)	20	1000	1000

2.2 实验设计

本文数学启发算法使用 Python 程序设计语言编程实现。算法步骤如下：读数据，使用 LH 方法获得一组初始解，迭代执行邻域选择和子问题求解，直到满足终止条件。子问题求解环节，使用 PuLP(https://github.com/coin-or/pulp)线性规划建模工具完成 SSCFLP 子问题模型建构，再调用 IBM CPLEX 12.6 优化器完成模型求解。为提升计算速度，算法在 PyPy7 (https://www.pypy.org)环境中运行。实验计算环境为：HP 桌面计算机，配置 Intel Core I7-6700 CPU 3.40-GHz 和 8GB 内存，Windows 10 操作系统。

案例测试过程如下。首先，设置参数 *mloops*，即连续未更新最好解循环数。每组案例使用参数依据经验设置：案例 OR1、OR2、OR3、H1 和 H2 设置为 10，案例 OR4、T2 设置为 50，案例 H3、H4 和 H5 设置为 20，其原理设置为 100。一般来说，参数 *psize* 数值越大，计算结果越稳定，所需计算时间也越长。其次，针对每个案例，使用数学启发算法计算 5 次，统计 5 个目标值的最小值、平均值、最大值和标准差，以及平均计算时

间。平均值与案例目标值下界之差反映算法求解质量，标准差用于评估算法的稳健性，计算时间用于评价算法的计算效率。

2.3 计算结果

使用本文数学启发算法对 5 个案例集中 272 个案例进行求解，结果统计如表 2。每个案例计算 5 次，统计目标值平均值和计算时间平均值。再计算目标值与案例目标值下界的差异值 Gap，Gap 值越小说明算法求解质量越高。当案例最优解已知时，Gap=(目标值平均值-最优解目标值)/最优解目标值*100%；当案例最优解未知时，Gap=(目标值平均值-目标值下界)/目标值下界*100%。表中，“Exact”栏列出精确求解获得最优解的数量“#Opt”以及案例平均求解时间，来自 CPLEX 分支切割算法或改进割平面算法计算结果。“MH”栏列出本文数学启发算法求解结果的 Gap 平均值和计算时间平均值；“KS”栏为核搜索算法(Guastaroba & Speranza 2014)结果统计；“HMEH” 栏为基于超图的多交换启发算法(Tran et al. 2017)求解结果统计；“CM”栏为“廊道”方法(Caserta & Voß 2020)结果统计。注意，表中案例求解质量可以直接对比，但计算时间不宜直接对比。

表 2 案例计算结果统计

案例	分组	Exact		MH		KS		HMEH		CM	
		#opt	Time/s	Gap/%	Time/s	Gap/%	Time/s	Gap/%	Time/s	Gap/%	Time/s
OR-Lib	OR1	8/8	0.05	0.00	1.22	0.00	0.29	-	-	-	-
	OR2	8/8	0.06	0.00	2.23	0.00	0.39	-	-	-	-
	OR3	8/8	0.08	0.00	1.56	0.00	0.62	-	-	-	-
	OR4	12/12	112.44	0.01	149.06	0.00	34.67	0.04	42.67	0.00	43
Holm.	H1	12/12	0.20	0.00	1.13	0.00	0.32	0.00	0.42	-	-
	H2	12/12	0.34	0.02	2.13	0.00	0.38			-	-
	H3	16/16	2.61	0.00	8.64	0.00	2.43	0.00	4.08	-	-
	H4	15/15	0.67	0.00	3.88	0.00	0.54	0.00	1.08	-	-
	H5	16/16	5.29	0.00	15.59	0.00	2.32	0.00	15.53	-	-
Yang	Y1	5/5	51.00	0.00	76.31	0.00	411.28	-	-	-	-
	Y2	5/5	1261.82	0.01	27.78	0.00	1640.42	-	-	-	-
	Y3	5/5	65.63	0.04	120.58	0.00	597.06	-	-	-	-
	Y4	5/5	228.01	0.09	232.79	0.00	1409.11	-	-	-	-
TB4	G1	15/15	676	0.03	86.55	-	-	-	-	-	-
	G2	13/15	4036	0.07	47.83	-	-	-	-	-	-
	G3	10/15	14617	0.07	61.59	-	-	-	-	-	-
Tebd1	T1	13/20	3722	0.15	60.88	0.56	2206.96	0.54	428.03	0.23	807
	T2	20/20	47	0.00	87.12	0.00	334.71	0.01	1159.33	0.00	29
	T3	8/20	2017	0.27	158.69	0.66	4190.28	0.52	2982.72	0.36	1024
	T4	4/20	7744	0.47	341.18	0.90	5244.69	0.82	4992.14	0.78	912
	T5	0/20	8275	0.57	345.58	1.07	6533.15	1.10	8582.74	1.11	932

表 2 中，案例 OR1、OR2 和 OR3 由本文作者使用 CPLEX 12.6 求解，获得全部最优解；案例 OR4、T1、T2、T3、T4 和 T5 是使用 CPLEX12.2 的计算结果(Guastaroba et al. 2014)；案例集 Holmberg、Yang 和 TB4 计算结果来自改进割平面算法(Gadegaard et al. 2018)。据文献 Gadegaard et al. (2018)，改进割平面算法求解 TB4 案例，获得 41 个案例的最优解。作者发现，其中 3 个案例的最优解有错误，因此表 1 中案例集 TB4 共有 38 个最优解。案例集 Tebd1 的最优解目标值、最优解案例数量和计算时间引自文献 Caserta

& Voß (2020)。表中可见，精确算法能够高效地求解规模较小的案例，如大部分 OR-Library、Holmberg 案例。然而，案例集 Yang、TB4 和 Tbed1 中多数案例计算时间很长。据文献 Yang et al.(2014)，使用 CPLEX 求解 Yang 案例，每个案例计算时间限制为 50000 秒，20 个案例中 14 个案例找不到最优解。据文献 Caserta & Voß (2020)，使用 CPLEX 求 Tbed1 案例，100 个案例有 45 个获得最优解。综上，精确算法求解中大规模 SSCFLP 案例存在困难，需要设计启发式算法求解这一问题。

从表 2 种可以发现：(1)对于较小规模案例(OR1、OR2 和 OR3)，启发式算法中，核搜索算法表现最好，但 CPLEX 分支切割算法求解更高效。(2)随案例规模增大，对于 Holmberg 和 OR4 案例，核搜索算法表现最好，在求解质量与计算时间方面与割平面算法差异不明显。(3)对于 Yang 案例，CPLEX 精确求解困难，割平面方法大幅降低了求解时间，且获得最优解；核搜索算法能够获得最优解，但计算时间较长；数学启发能够在短时间内获得高质量解。(4)对于 TB4 案例，CPLEX 精确求解困难，割平面方法大幅降低了求解时间，获得大多数案例最优解，而数学启发能够在短时间内获得高质量解。(5)对于更大规模案例集 Tbed1，割平面算法的可行性尚未验证；CPLEX 能够高效求解部分案例，如 T2 案例和 T1 中少数案例，但对于其他案例，精确求解效率偏低，或者难以精确求解；启发式算法核搜索和大邻域搜索算法能较好地求解 Tbed1 中的大规模案例，本文数学启发算法性能更优，特别是对于较大规模案例 T4 和 T5，求解质量明显提升，且计算时间大幅度下降。综上分析，数学启发算法在求解大规模案例(如 T3、T4、T5)或者较困难案例(如 TB4)方面具有明显的优势。

使用数学启发算法进行案例测试过程中，作者发现现有文献中存在某些错误。首先，TB4 案例集中，3 个案例(50-200-2-4、50-200-2-5 和 60-200-2-5)的目标值小于文献 Gadegaard et al. (2018)提供的最优目标值。使用 CPLEX 重新计算案例模型，结果表明该文献提供的案例最优解错误。第二，Tbed1 案例集中，部分案例目标值小于文献 Guastaroba et al. (2014)提供的最优目标值，应采用文献 Caserta & Voß (2020)提供了新的 CPLEX 计算结果。第三，文献 Tran et al. (2017)更新的最好解中，个别案例目标值显著小于案例最优目标值，如 Tbed1 案例 i300_7、i300_10 和 i500_13。另外，作者使用 CPLEX 成功求解个别案例，更新了最优解，如 Tbed1 案例 i300_6、i300_7、i500_6、i500_7、i500_9、i500_10、i500_13、i500_15、i700_14 和 i700_20。表 3 为部分案例目标值上界、下界和最优值的更新或更正，标注星号的为最优值。

表 4 汇总常见算法发现最优解的数量。表中，案例已知最优解数量“#Opt”来自文献和作者计算结果，已统计表 3 所列案例。已知最好解引自文献 Yang et al. (2012)、Guastaroba et al.(2014)、Tran et al. (2017)和 Gadegaard et al. (2018)。统计时，剔除了文献 Tran et al. (2017)和 Gadegaard et al. (2018)中有错误的案例。表中，MH、CS、CS2、KS、HMEH、CM 分别表示数学启发算法、割平面算法、改进割平面算法、核搜索算法和廊道方法。对于 20 个 Yang 案例，CPLEX 求解困难，割平面算法(Yang et al. 2012, Gadegaard et al. 2018)能够发现全部案例的最优解，核搜索启发算法和本文数学启发算法均发现了 18 个最优解。对于 45 个 TB4 案例，改进割平面算法(Gadegaard et al. 2018) 获得 38 个案

例的最优解(已剔除 3 个有错误的案例)，本文数学启发算法发现 33 个案例的最优解、更新 4 个案例的最好解，计算时间大幅降低。对于 100 个大规模案例 Tbed1，CPLEX 计算结果共发现 55 个案例的最优解，但计算时间偏长；核搜索启发算法获得 40 个案例最优解，HMEH 算法发现 28 个案例最优解，廊道方法发现 44 个最优解；本文数学启发算法获得 41 案例最优解，计算时间大幅降低。

表 3 部分案例 CPLEX 计算结果

案例集	案例	目标下界	目标上界	备注
TB4	50-200-2-4	25951.63	25955	更正(Gadegaard et al. 2018)
TB4	50-200-2-5	25326.5	25329	更正(Gadegaard et al. 2018)
TB4	50-300-2-5	37138.1	37142	更正(Gadegaard et al. 2018)
Tbed1	i300_6	11326.43*	11326.43	更新(Caserta & Voß 2020)
Tbed1	i300_7	11470.31*	11470.31	更新(Caserta & Voß 2020)
Tbed1	i500_6	15853.35*	15853.35	更新(Caserta & Voß 2020)
Tbed1	i500_7	16205.15*	16205.15	更新(Caserta & Voß 2020)
Tbed1	i500_9	16399.40*	16399.4	更新(Caserta & Voß 2020)
Tbed1	i500_10	15886.54*	15886.54	更新(Caserta & Voß 2020)
Tbed1	i500_13	13715.96*	13715.96	更新(Caserta & Voß 2020)
Tbed1	i500_15	13947.12*	13947.12	更新(Caserta & Voß 2020)
Tbed1	i700_14	17383.87*	17383.87	更新(Caserta & Voß 2020)
Tbed1	i700_20	15492.02*	15492.02	更新(Caserta & Voß 2020)

表 4 常见算法发现最优解的案例数量

案例集	分组	#案例	#Opt	CPLEX	MH	CS	CS2	KS	HMEH	CM
OR-Lib	OR4	12	12	12	11	-	-	12	-	12
Yang	Y1	5	5	3	5	5	5	5	-	-
	Y2	5	5	1	4	5	5	4	-	-
	Y3	5	5	1	5	5	5	4	-	-
	Y4	5	5	1	4	5	5	5	-	-
TB4	G1	15	15	-	12	-	15	-	-	-
	G2	15	13	-	7	-	13	-	-	-
	G3	15	10	-	7	-	10	-	-	-
Tebd1	T1	20	15	15	12	-	-	12	10	12
	T2	20	20	20	17	-	-	20	14	20
	T3	20	14	14	7	-	-	6	4	8
	T4	20	6	6	5	-	-	2	0	4
	T5	20	0	0	0	-	-	0	0	0

对于 53 个最优解未知的 TB4 和 Tbed1 案例，本文更新了其中 36 个案例的已知最好解，见表 5。表中包括 4 个 TB4 案例和 32 个 Tbed1 案例。TB4 案例目标值下界和已知最好解目标值由改进割平面算法(Gadegaard et al. 2018))提供；Tbed1 案例目标值下界来自 CPLEX 求解，已知最好解目标值来自 CPLEX、核搜索、HMEH 或廊道方法。新最好解是本文算法求解结果。Tbed1 案例的最好解处于持续更新中，HMEH 算法、廊道算法更新了部分案例最好解，本文算法进一步更新了其中 32 个案例的最好解。与原有最

好解相比，这些案例目标值平均改进了 0.48%。其中，案例 60-300-2-4、i700_1、i700_2、i700_4、i1000_1、i1000_2、i1000_3、i1000_4、i1000_5 和 i1000_10 的改进幅度明显，处于 0.75%~3.55%区间。

表 5 更新已知最好解目标值

案例集	案例	下界	已知最好解	新最好解
TB4	60-300-2-1	34858.5	34861	34860
TB4	60-300-2-2	36543.5	36742	36551
TB4	60-300-2-3	34876.2	34884	34878
TB4	60-300-2-4	34817.6	36057	34821
Tbed1	i300_2	16059.34	16140.*00*	16135.82
Tbed1	i300_3	15606.10	15687.38	15666.23
Tbed1	i300_4	18143.89	18312.60	18255.10
Tbed1	i300_5	18191.11	18315.44	18291.05
Tbed1	i500_1	26566.69	26824.08	26731.63
Tbed1	i500_3	28067.68	28362.79	28284.36
Tbed1	i500_4	28268.36	28518.40	28489.69
Tbed1	i700_1	37054.60	37751.08	37343.34
Tbed1	i700_2	34488.56	35076.83	34817.44
Tbed1	i700_3	34485.24	34977.47	34759.07
Tbed1	i700_4	38260.98	38860.34	38534.35
Tbed1	i700_7	21297.30	21437.82	21433.56
Tbed1	i700_8	20659.96	20823.75	20820.70
Tbed1	i700_10	22055.41	22274.57	22210.50
Tbed1	i700_11	17120.15	17189.64	17188.47
Tbed1	i700_12	18135.97	18232.53	18201.06
Tbed1	i1000_1	49681.02	50734.33	50104.98
Tbed1	i1000_2	50842.16	51677.00	51277.80
Tbed1	i1000_3	47362.62	48141.82	47737.73
Tbed1	i1000_4	49029.12	49910.85	49408.86
Tbed1	i1000_5	50971.44	51824.38	51166.18
Tbed1	i1000_6	27823.84	28051.58	28043.73
Tbed1	i1000_7	27252.32	27521.50	27412.21
Tbed1	i1000_8	27375.37	27638.39	27543.01
Tbed1	i1000_9	26857.09	27127.70	26992.81
Tbed1	i1000_10	27186.99	27469.49	27242.63
Tbed1	i1000_11	22180.33	22297.32	22247.61
Tbed1	i1000_12	22160.39	22231.34	22231.18
Tbed1	i1000_13	22657.09	22768.69	22745.61
Tbed1	i1000_15	22629.44	22706.59	22704.67
Tbed1	i1000_16	21331.81	21389.82	21365.65
Tbed1	i1000_20	21560.86	21618.06	21601.55

2.4 算法机制分析

作者进一步验证了算法机制或参数设置对计算性能的影响。首先，初始解对计算结果影响显著。作者设计三种初始解生成方法：LH、线性松弛启发和 ADD 方法。线性松弛启发方法是将 SSCFLP 的指派变量修改为连续变量，求解模型，再修复多源指派为单源指派。ADD 方法过程如下：随机选择一个设施，进行贪心指派；以贪心的方式选择设施并进行指派，直到所有需求得到满足。比较发现：使用三种方法整体差异不大，线性松弛启略优于 LH 算法，LH 算法优于 ADD 方法。其次，在大邻域选择时，测试是否使用设施排除机制。测试结果发现，两者在求解质量和计算时间方面均无明显差异。第三，使用群解搜索取代单解搜索，即维持一组精英解，算法随机选择一个精英解作为当前解进行搜索。测试发现，群解搜索略好于单解搜索，但计算时间有明显的增加。

3 结论及讨论

本文设计了数学启发算法求解 SSCFLP 案例，并使用 5 个案例集测试了算法性能。算法原理如下：从当前解中选择一个局部区域，建立该局部区域的子问题模型，使用 MIP 优化器求解，然后更新当前解。迭代执行这一过程，直到当前解难以更新。272 个案例测试表明：本文数学启发算法共发现 191 个案例的最优解，更新 36 个案例的已知最好解；在困难案例(案例集 Yang 和 TB4)和大规模案例(案例集 Tbed1)上，解的质量整体上最优，计算时间明显降低。

从算法原理方法的角度，本文数学启发算法属于超大邻域搜索(VLNS)算法。本文算法与 HMEH 算法(Tran et al. 2017)类似，均是根据供需空间分布确定一个超大邻域，在大邻域空间中进行邻域搜索，但也存在两点不同之处。首先，邻域空间定义不同。HMEH 算法选择的邻域空间涉及 15%~20%的候选设施数量，不考虑当前使用设施数量；本文算法从当前解中选择 7 个左右的设施，将这些设施所覆盖的范围作为邻域空间。对于超大规模案例 T4 和 T5，HMEH 算法定义的邻域空间巨大，而本文算法定义的邻域空间相对较小。第二，两者搜索方法不同。HMEH 算法基于动态构建的超图，在并行处理框架中，使用“前向选择”和“后向选择”方法尝试改进当前解；而本文方法是构建邻域空间内的子问题数学模型，使用 MIP 求解器求解，进而更新当前解。因本文将邻域空间大小定义在一个可以接受的范围内，对应的子问题可以较为高效地求解，也使算法效率较高。算法测试表明，本文数学启发算法在求解质量和计算效率方法均优于 HMEH 算法。

SSCFLP 算法具有几个明显的特征。第一，拉格朗日松弛方法在不同案例集上的求解质量差异较大，但因其计算速度快，常作为复杂算法的初始解(Ahuja et al. 2004; Tran et al. 2017)，或者其他算法的构件(Holmberg et al. 1999; Chen & Ting 2008)。第二，除 HMEH 外，高性能 SSCFLP 算法依赖于 MIP 优化器。割平面(Yang et al. 2012)、改进割平面(Gadegaard et al. 2018)、核搜索(Guastaroba et al. 2014)、廊道方法(Caserta & Voß 2020)以及本文数学启发均需要 MIP 优化器。超大邻域搜索有可能较好地求解 SSCFLP，但超大邻域涉及多个设施，又需要兼顾几十甚至上百个需求点的指派，组合数量庞大，搜索计算复杂度极高，而 MIP 优化器在超大规模邻域搜索方面具有优势。第三，针对 SSCFLP 的

群算法报道较少，蚁群算法(Chen & Ting 2008) 表现较好，但该算法未在大规模案例上进行测试。群算法较少的可能原因：区位选择与需求指派的解空间过于庞大，两个可行解差异很大，交叉操作难以形成可行解，可行性修复也很困难，从而造成演化过程非常缓慢；类似地，群智能算法需要庞大的种群才能较为有效地探索解空间。

应到注意到，本文算法仍有需要完善的地方。对于小规模案例(如 OR-Lib 和 Holmberg 案例)，数学启发算法缺乏优势，虽能获得几乎所有案例的最优解，但算法计算时间明显过长。原因在于：这些案例相对容易求解，改进分枝定界(Holmberg et al. 1999)、核搜索、HMEH、割平面算法均表现最好，直接使用 CPLEX 等优化器也能高效地求解这些案例。因此，对于小规模案例，数学启发算法迭代求解 MIP 模型显得冗余，较大幅地增加了求解时间。未来进一步的研究包括：分析 SSCFLP 案例模型的求解难度，判断直接模型求解还是采用数学启发方法求解；分析邻域空间大小对求解质量和计算效率的影响，探索更优的邻域定义方法；验证数学启发算法对于 SSCFLP 变种问题的有效性，如限制设施数量、要求设施服务区空间连续等情形。

参考文献

[1] Agar M, Salhi S. Lagrangean heuristics applied to a variety of large capacitated plant location problems. J. Oper. Res. Soc., 1998, 49:1072–1084.

[2] Ahuja R K, Orlin J B, Sharma D. Multi-exchange neighborhood search algorithms for the capacitated minimum spanning tree problem. Math. Programming, 2001, 91:71–97.

[3] Avella P, Boccia M. A cutting plane algorithm for the capacitated facility location problem. Computational Optimization and Applications, 2009, 43 (1): 39-65.

[4] Barceló J, Casanovas J. A heuristic Lagrangean algorithm for the capacitated plant location problem. Eur J Oper Res. , 1984, 15(2):212–226

[5] Barceló J, Fernández E, Jörnsten K. Computational results from a new Lagrangean relaxation algorithm for the capacitated plant location problem. Eur J Oper Res, 1991, 53(1):38–45.

[6] Basu S, Sharma M & Ghosh P S. Metaheuristic applications on discrete facility location problems: a survey. OPSEARCH, 2015, 52, 530–561.

[7] Beasley J E. Lagrangian heuristics for location problems. Eur. J. Oper. Res., 1993, 65:383–399.

[8] Beasley J E. OR-Library: Distributing test problems by electronic mail. J. Oper. Res. Soc., 1990. 41:1069–1072.

[9] Boccia M, Sforza A, Sterle C, Vasilyev I. A cut and branch approach for the capacitated p-median problem based on Fenchel cutting planes. J Math Modell Algorithms, 2008, 7(1):43–58.

[10] Caserta M & Voß S. A general corridor method-based approach for capacitated facility location. International Journal of Production Research, 2020, 58:13: 3855-3880.

[11] Chen C.H., C.J. Ting. Combining lagrangian heuristic and ant colony system to solve the single source capacitated facility location problem. Transportation Research Part E: Logistics and Transportation Review, 2008, 44 (6): 1099-1122.

[12] Contreras I A, Díaz J A. Scatter search for the single source capacitated facility location problem. Annals of Operations Research, 2008, 157 (1):73-89.

[13] Cornuejols G, Sridharan R, Thizy J. A comparison of heuristics and relaxations for the capacitated plant location problem. Eur J Oper Res, 1991, 50(3):280–297.

[14] Cortinhal M J, Captivo M E. Upper and lower bounds for the single source capacitated location problem. European Journal of Operational Research, 2003, 151(2): 333-351.

[15] Delmaire H, Díaz J A, Fernández E, Ortega M. Reactive GRASP and tabu search based heuristics for the single source capacitated plant location problem. Infor-Information Systems and Operational Research, 1999,37 (3): 194-225.

[16] Díaz J, Fernández E. A branch-and-price algorithm for the single source capacitated plant location problem. J Oper Res Soc, 2002, 53:728–740.

[17] Filho V J M F, Galvao R D. A tabu search heuristic for the concentrator location problem. Location Science, 1998, 6 (1–4), 189–209.

[18] Gadegaard S L, Klose A, Nielsen L R. An improved cut-and-solve algorithm for the single-source capacitated facility location problem. EURO J Comput Optim, 2018, 6, 1–27.

[19] Guastaroba G, Speranza M G. A heuristic for BILP problems: The Single Source Capacitated Facility Location Problem, European Journal of Operational Research, 2014, 238 (2): 438-450.

[20] Hindi H, Pieńkosz K. Efficient solution of large scale, single-source, capacitated plant location problems. J. Oper. Res. Soc., 1999, 50:268–274

[21] Holmberg K, Rönnqvist M, Yuan D. An exact algorithm for the capacitated facility location problems with single sourcing. European Journal of Operational Research, 1999, 113 (3): 544-559.

[22] Holt J, Ronnqvist M, Tragantalerngsak S. A repeated matching heuristic for the single-source capacitated facility location problem. Eur. J. Oper. Res., 1999, 116:51–68.

[23] Klincewicz J., Luss H. A Lagrangean relaxation heuristic for capacitated facility location with single-source constraints. J. Oper. Res. Soc., 1986, 37:495–500.

[24] Klose A, Görtz S. A branch-and-price algorithm for the capacitated facility location problem. Eur J Oper Res, 2007, 179(3):1109–1125.

[25] Neebe A W, Rao M R. An algorithm for the fixed-charge assigning users to sources problem The Journal of the Operational Research Society, 1983, 34:1107–1113.

[26] Oliveira Ó, Matos T, Gamboa D. A RAMP Algorithm for Large-Scale Single Source Capacitated Facility Location Problems. In: Matsatsinis N, Marinakis Y, Pardalos P. (eds)

Learning and Intelligent Optimization. LION 2019. Lecture Notes in Computer Science, 2020, vol 11968, Springer, Cham.

[27] Rönnqvist M, Tragantalerngsak S, Holt J. A repeated matching heuristic for the single-source capacitated facility location problem. Eur J Oper Res, 1999, 116(1):51–68.

[28] Sridharan R. A Lagrangian heuristic for the capacitated plant location problem with single source constraints. Eur J Oper Res, 1993, 66(3):305–312.

[29] Tran T H, Scaparra M P, O'Hanley J R. A hypergraph multi-exchange heuristic for the single-source capacitated facility location problem, European Journal of Operational Research, 2017, 263(1): 173-187.

[30] Ulukan Z, Demircioğlu E. A Survey of Discrete Facility Location Problems. International Journal of Industrial and Manufacturing Engineering, 2015, 9(7): 2487-2492.

[31] Yang Z, Chu F, Chen H. A cut-and-solve based algorithm for the single-source capacitated facility location problem. European Journal of Operational Research, 2012, 221 (3): 521-532.

A matheuristic algorithm for the single-source capacitated facility location problem

Abstract: This article proposes a matheuristic algorithm for the classical single-source facility location problem (SSCFLP). The algorithm starts from an initial solution, and then iteratively improves the solution by searching the large-scale neighborhoods. The initial solution is generated by a Lagrangian heuristic and the large neighborhoods are explored by solving sub-problems. The performance of the algorithm was tested on 5 set of benchmark instances. Experimentation showed that the instances could be solved effectively and efficiently. For the largest set of instances, the proposed matheuristic algorithm performs better than existing methods in terms of the solution quality, the computational time, the number of optimal solutions and the number of better solutions.

Keywords: single-source capacitated facility location problem, matheuristic algorithm; Large-scale neighborhood; sub-problem.